

REQUIREMENT ANALYSIS

Problem Description

As it is known, in all educational institutions, it is necessary to store some data or make a transaction about the students, courses, lecturers etc. However, it can be hard to do these on paper, especially in large systems like universities.

That's why, A system including all the rules inside is required to store the information of an actor and inform all the system (other actors) so that they (students, lecturers, head of departments and faculties, etc.) can follow the flow of the system.

Vision

Designing and implementing a simulation for customers to simulate course registration system. With the help of the simulation, customers can find out how the course registration process works and observe the problems that may occur in during process.

Scope

- This simulation provides statistical data about courses and students in the end of the registration process.
- This simulation provides information about specific problems during the registration process.

Iteration Plan

First Iteration: In the first iteration, we will spend our energy and focus into documentation phase mostly. We will hold several meetings to agree on core functionalities on the general concept of the project. After, we will assign small groups for the analysis and design phase. We will start preparing the requirement analysis document. After that, we will work on design phase. When we think that the documentation is sufficient for starting the coding phase, we will start coding according to the analysis and design phase implications.

Second Iteration: In the second iteration, we will spend our energy into completing all the requirements. For this purpose, we will first determine the tasks by considering what we did in first iteration and what we got as a feedback in the demo session. Then, we will distribute these tasks among us. The tasks are updating RAD by adding 'Problem Description' field, updating Glossary, updating Object Domain Model and adding 'System Sequence Diagram; updating DCD by updating the attributes, methods and connections between the classes; updating DSD by simplifying a bit and updating according to changes; adding loggers and adding unit tests.

Glossary of Term

Student: Main actor of the system.

Transcript: Detailed and approved record of students grades that has been generated by university.

Registration Process: Advisor-controlled course selection process of student's enrollment.

Curriculum: The subjects comprising a course of study in a university.

Semester: A half year term in a university.

Prerequisite Tree: It is the model that contains the pre-conditional relations between the courses in the curriculum. In this project we will simulate prerequisite tree for the students who registered in 2020 and later.

Advisor: University academic member who is approving the students' academic program.

Quota: A limited number of students that can enroll to specific course.

Course: Syllabus item offered by the university.

Credit: The term that measures the numerical value of a course based on hours in a week or the content of the course, used for selecting some of the elective courses and graduation.

Unit Test: Used to verify that a single piece of code is working as expected.

TE Course: Technical Electives are courses (in our case computer science) oriented toward the design or use of computers.

Graduation Project: is a formal assignment chosen by a student or small group of students on a topic related to interested field.

Functional Requirements

- Certain number of students for each semester should be generated randomly with the given fields:

Student ID, GPA, Name, Surname and Transcript

- Registration process should be simulated for each student that are generated randomly.
- Problems that may occur regarding course registration should be checked.
- Successfully registered courses should be recorded on the student's transcript.
- If a student fails to register a course the reason behind it should be written to its json file.
 - o to register a course, student must pass its pre-requisites.
 - o to register a course, that course's quota must not be exceeded.
 - o Student cannot enroll courses from an upper semester.
 - o to register Graduation Project 1, student must have 165 credits
 - o to register TE courses, student must have 155 credits
 - o A student cannot take more than 10 course in one semester.
- An output file containing general statistics about the course registration process should be created on a departmental basis

Non-Functional Requirements

- Simulation should be implemented using Java programming language.
- Considering the addition of different iterations, the program has been developed in accordance with the iterative progress principles.
- The program will use json format both inputs and outputs files.
- The simulation will not be supported by user interface. All outputs will be accessed from command-line and json files.
- Outputs and logs of the system must be proper manner and to be understandable clearly.
- All properties must be in input json file.
- Unit testing will be done on several important classes with using JUnit

Use-case

Scenario: Registration to Course

Actors: Student, Advisor, System

- 1- The student send request for registration processes with semester info.
- 2- The student will be assigned to the advisor.
- 3- The student sends the request course list to the advisor.
- 4- The advisor evaluates the submitted request course list, considering the prerequisites and other conditions.
- 5- A transcript will be created for each student and the course information will be kept.
- 6- Then, the registration process for the related semester is started for the related student.
- 7- The student's enrollment output will be saved in a file together with the transcript before and after the enrollment.
- 8- In the output file it will output general statistics about the registration issues for the partition.

Alternatives

4.a - Selected course's quota is full therefore student cannot take that course. Advisor won't approve that course and drops that course from student's course list.

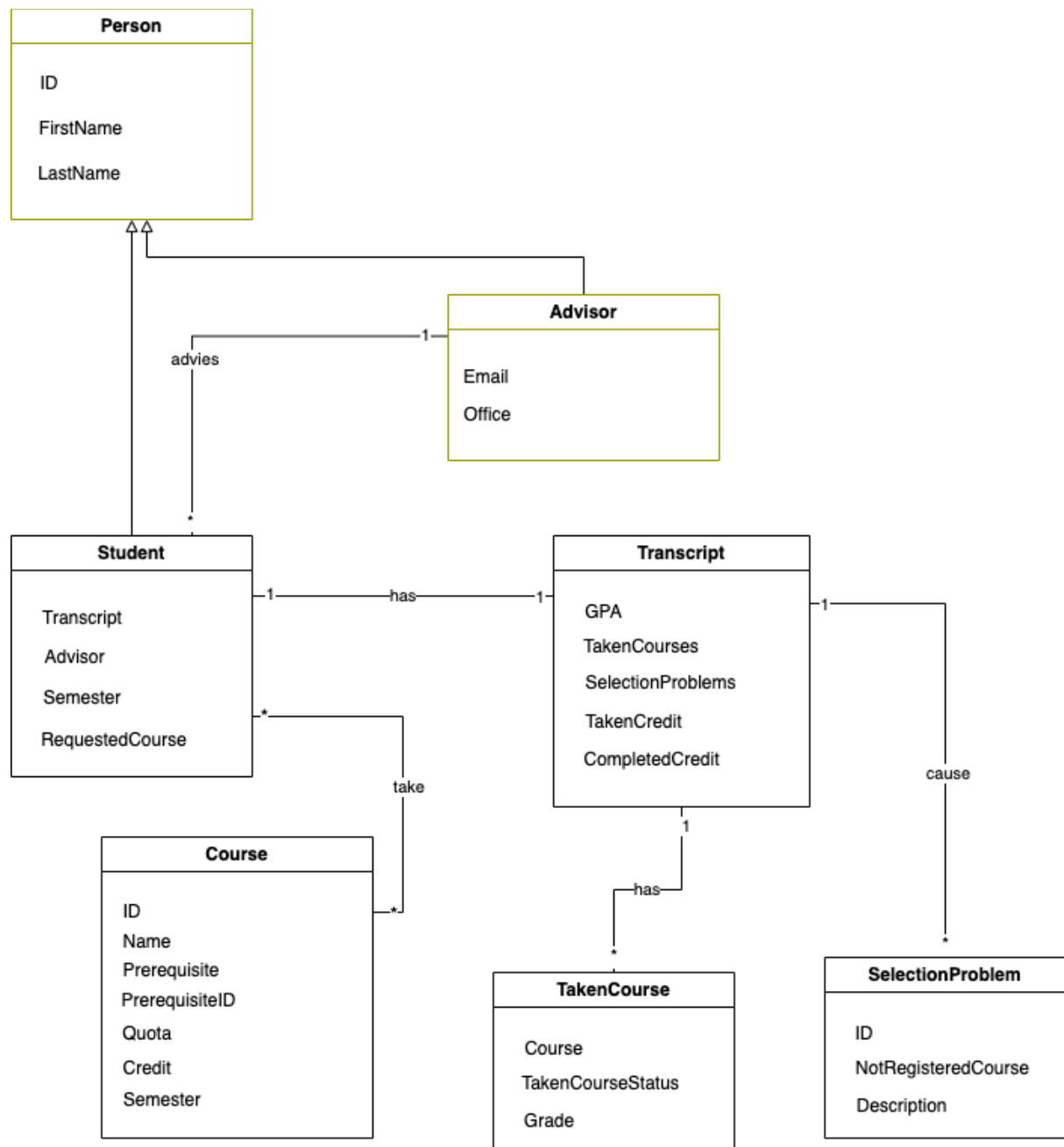
4.b - Student haven't passed pre-requisite(s) of selected course. Advisor won't approve that course and drops that course from student's course list.

4.b.1 - In this case, the system checks whether the student can take the pre-requisite(s) course.

4.c- Students must complete 165 credits to take Graduation Project 1 course. If not completed, Advisor won't approve that course and drops that course from student's course list.

4.d Students must complete 155 credits to take TE courses. If not completed, Advisor won't approve that course, drops that course from student's course list.

Object Domain Model



SYSTEM SEQUENCE DIAGRAM

